Modeling Perceptual Color Differences by Local Metric Learning

Michaël Perrot, Amaury Habrard, Damien Muselet, Marc Sebban

Modeling Perceptual Color Differences by Local Metric Learning

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban

LaHC, UMR CNRS 5516, Université Jean-Monnet, F-42000, Saint-Étienne, France
{michael.perrot,amaury.habrard,damien.muselet,marc.sebban}@univ-st-etienne.fr

Abstract. Having perceptual differences between scene colors is key in many computer vision applications such as image segmentation or visual salient region detection. Nevertheless, most of the times, we only have access to the rendered image colors, without any means to go back to the true scene colors. The main existing approaches propose either to compute a perceptual distance between the rendered image colors, or to estimate the scene colors from the rendered image colors and then to evaluate perceptual distances. However the first approach provides distances that can be far from the scene color differences while the second requires the knowledge of the acquisition conditions that are unavailable for most of the applications. In this paper, we design a new local Mahalanobis-like metric learning algorithm that aims at approximating a perceptual scene color difference that is invariant to the acquisition conditions and computed only from rendered image colors. Using the theoretical framework of uniform stability, we provide consistency guarantees on the learned model. Moreover, our experimental evaluation shows its great ability (i) to generalize to new colors and devices and (ii) to deal with segmentation tasks.

Keywords: Color difference, Metric learning, Uniform color space

1 Introduction

In computer vision, the evaluation of color differences is required for many applications. For example, in image segmentation, the basic idea is to merge two neighbor pixels in the same region if the difference between their colors is "small" and to split them into different regions otherwise [4]. Likewise, for visual salient region detection, the color difference between one pixel and its neighborhood is also the main used information [1], as well as for edge and corner detection [27, 28]. On the other hand, in order to evaluate the quality of color images, Xue et al. have shown that the pixel-wise mean square difference between the original and distorted image provides very good results [36]. As a last example, the orientation of gradient which is the most widely used feature for image description (SIFT [16], HOG [7]) is evaluated as the ratio between vertical and horizontal differences.

Depending on the application requirement, the used color difference may have different properties. For material edge detection, it has to be robust to local photometric variations such as highlights or shadows [28]. For gradient-based color descriptors, it has to be robust to acquisition condition variations [6, 20] or discriminative [27]. For most applications and especially for visual saliency detection [1], image segmentation [4] or image quality assessment [36], the color difference has to be above all perceptual, i.e. proportional to the color difference perceived by human observers. In the computer vision community, some color spaces such as CIELAB or CIELUV are known to be closer to the human perception of colors than RGB. It means that distances evaluated in these spaces are more perceptual than distances in the classical RGB spaces (which are known to be non uniform). Thus, by moving from RGB to one of these spaces with a default transformation [23, 24], the results of many applications have improved [1, 2, 4, 11, 18]. Nevertheless, it is important to know that this default approach provides a perceptual distance between the colors in the rendered image (called image-wise color distance) and not between the colors as they appear to a human observer looking at the real scene (called scene-wise color distance). The transformation from the scene colors to the image rendered colors is a succession of non-linear transformations which are device specific (white balance, gamma correction, demosaicing, compression, ...). For some applications such as image quality assessment, it is required to use the image-wise color distances since only the rendered image colors need to be compared, whatever the scene colors. But for a lot of other applications such as image segmentation, saliency detection, ..., we claim that a scene-wise perceptual color distance should be used. Indeed, in these cases, the aim is to be able to evaluate distances as they would have been perceived by a human observing the scene and not after the camera transformations. Some solutions exist [12] to get back to scene colors from RGB camera outputs but they require calibrated acquisition conditions (known illumination, known sensor sensitivities, RAW data available,...).

In this paper we propose a method to estimate scene-wise color distances from non calibrated rendered image colors. Furthermore, we go a step further towards an invariant color distance. This invariance property means that, considering one image representing two color patches, the distance is predicting how much difference would have perceived a human observer looking at the two real patches under standard fixed viewing conditions, such as the ones recommended by the CIE (Commission Internationale de l'Eclairage) in the context of color difference assessment [22]. In other words, whatever the acquisition device or the illuminant, an invariant scene-wise distance should return stable values.

Since the acquisition condition variability is huge, rather than using models of invariance [6, 20] and models of acquisition devices [13, 34], we propose to automatically learn an invariant perceptual distance from training data. In this context, our objective is three-fold and takes the form of algorithmic, theoretical and practical contributions:

- First, we design a new metric learning algorithm [37] dedicated to approximate reference perceptual distances from the image rendered RGB space. It aims

at learning local Mahalanobis-like distances in order to capture the non linearity required to get a scene-wise perceptual color distance.

- Second, modeling the regions as a multinomial distribution and making use of the theoretical framework of uniform stability, we derive consistency guarantees on our algorithm that show how fast the empirical loss of our learned metric converges to its true generalization value.

- Lastly, to learn generalizable distances, we create a dataset of color patches that are acquired under a large range of acquisition conditions (different cameras, illuminations, viewpoints). We claim that this dataset [37] may play the role of benchmark for the computer vision community.

The rest of this paper is organized as follows: Section 2 is devoted to the presentation of the related work in color distances and metric learning. In Section 3, we present the experimental setup used to generate our dataset of images. Then, we introduce our new metric learning algorithm and perform a theoretical analysis. Finally, Section 4 is dedicated to the empirical evaluation of our algorithm. To tackle this task, we perform two kinds of experiments: first, we assess the capability of the learned metrics to generalize to new colors and devices; second, we evaluate their relevance in a segmentation application. We show that in both settings, our learned metrics outperform the state of the art.

2 Related Work

2.1 Perceptually uniform color distance

A large amount of work has been done by color scientists around perceptual color differences [31, 9, 22], where the required inputs of the proposed distances are either *reflectance spectra* or the *device-independent color components* CIE XYZ [31]. These features are obtained with particular devices such as spectrophotometer or photoelectric colorimeter [31]. It is known that neither the euclidean distance between reflectance spectra nor the euclidean distance between XYZ vectors are perceptual, i.e. these distances can be higher for two colors that look similar than for two colors that look different. Consequently, some color spaces such as CIELAB or CIELUV have been designed to be more perceptually uniform. In those spaces, specific color difference equations have been proposed to improve perceptual uniformity over the simple euclidean distance [9]. The ΔE_{00} [22] distance is one nice example of such a distance. It corresponds to the difference perceived by a human looking at the two considered colors under standard viewing conditions recommended by the CIE (illuminant D65, illuminance of 1000 lx, etc.).

However, it is worth noting that in most of the computer vision applications, the available information does not take the form of a reflectance spectra or some device-independent components, as assumed above. Indeed, the classical acquisition devices are cameras that use iterative complex transforms from the irradiance (amount of light) collected by each CCD sensor cell to the pixel intensity of the output image [13]. These device-dependent transforms are color filtering, white-balancing, gamma correction, demosaicing, compression, etc. [34]

which are designed to provide pleasant images and not to accurately measure colors. Consequently, the available RGB components in color images do not allow us to get back to the original spectra or XYZ components. To overcome this limitation, two main strategies have been suggested in the literature: either by applying a default transformation from RGB components to $L^*a^*b^*$ (CIELAB space) or $L^*u^*v^*$ (CIELUV space) assuming a given configuration, or by learning a coordinate transform to actual $L^*a^*b^*$ components under particular conditions.

Using default transformations A classical strategy consists in using a default transformation from the available RGB components to XYZ and then to $L^*a^*b^*$ or $L^*u^*v^*$ [1, 4, 11, 18]. This default transformation assumes an average gamma correction of 2.2 [23], color primaries close to ITU-R BT.709 [24] and D65 illuminant (Daylight). Finally, from the estimated $L^*a^*b^*$ or $L^*u^*v^*$ (denoted $\widehat{L^*a^*b^*}$ and $\widehat{L^*u^*v^*}$ respectively) of two pixels, one can make use of the euclidean distance. In the case of $L^*a^*b^*$, one can use $\widehat{L^*a^*b^*}$ to estimate more complex and accurate distances such as ΔE_{00} via its estimate $\widehat{\Delta E_{00}}$ ([22]), that will be used in our experimental study as a baseline. As discussed in the introduction, when using this approach, the provided color distance characterizes the difference between the colors in the rendered image after the camera transformations and is not related to the colors of the scene.

Learning coordinate transforms to $L^*a^*b^*$ For applications requiring the distances between the colors in the scene, the acquisition conditions are calibrated first and then the images are acquired under these particular conditions [14, 15]. Therefore, the camera position and the light color, intensity and positions are fixed and a set of images of different color patches are acquired. Meanwhile, under the same exact conditions, a colorimeter measures the actual $L^*a^*b^*$ components (in the scene) for each of these patches. In [15], they learn then the best transform from camera RGB to actual $L^*a^*b^*$ components with a neural network. In [14], they first apply the default transform presented before from camera RGB to $\widehat{L^*a^*b^*}$ and then learn a polynomial regression (until quadratic term) from the $\widehat{L^*a^*b^*}$ to the true $L^*a^*b^*$. However, it is worth mentioning that in both cases the learned transforms are accurate only under these acquisition conditions. Thus, these approaches can not be applied on most of the computer vision applications where such an information is unavailable.

From our knowledge, no previous work has both underlined and answered the problem of the approximations that are made during the estimation of the $L^*a^*b^*$ components in the very frequent case of uncalibrated acquisitions. The standard principle consisting in applying a default transform leads to distances that are only coarsely perceptual with respect to the scene colors. We will see in the rest of this paper that rather than sequentially moving from space to space with inaccurate transforms, a better way consists in learning a perceptual metric directly in the image rendered RGB space. This is a matter of metric learning for which we present a short survey in the next section.

2.2 Metric learning

Metric learning (see [3] for a survey) arises from the necessity for a lot of applications to accurately compare examples. The underlying idea is to define application dependent metrics which are able to capture the idiosyncrasies of the data at hand. Most of the existing work in metric learning is focused on learning a Mahalanobis-like distance of the form $d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}')}$, where $\mathbf{M}$ is a positive semi-definite (PSD) matrix to optimize. Note that using a Cholesky decomposition of $\mathbf{M}$, the Malahanobis distance can be seen as a Euclidean distance computed after applying a learned data linear projection.

The work of [32] where the authors maximize the distance between dissimilar points while maintaining a small distance between similar points has been pioneering in this field. Following this idea, Weinberger and Saul [29] propose to learn a PSD matrix dedicated to improve the k-nearest neighbors algorithm. To do so, they force their metric to respect local constraints. Given triplets (z_i, z_j, z_k) where z_j and z_k belong to the neighborhood of z_i, z_i and z_j being of the same class, and z_k being of opposite class, the constraints impose that z_i should be closer to z_j than to z_k with a margin ε. To overcome the PSD constraint, which requires a costly projection of $\mathbf{M}$ onto the cone of PSD matrices, Davis et al. [8] optimize a Bregman divergence under some proximity constraints between pairs of points. The underlying idea is to learn $\mathbf{M}$ such that it remains close to a matrix $\mathbf{M_0}$ defined a-priori. If the Bregman divergence is finite, the authors show that $\mathbf{M}$ is guaranteed to be PSD.

An important limitation of learning a unique global metric such as a Mahalanobis distance comes from the fact that no information about the structure of the input space is taken into account. Moreover, since a Mahalanobis distance boils down to projecting the data into a new space via a linear transformation, it does not allow us to capture non linearity. Learning local metrics is one possible way to deal with these two issues[1]. In [30], the authors propose a local version of [29], where a clustering is performed as a preprocess and then a metric is learned for each cluster. In [26], Wang et al. optimize a combination of metric bases that are learned for some anchor points defined as the means of clusters constructed, for example, by the K-Means algorithm. Other local metric learning algorithms have been recently proposed, only in a classification setting, such as [33] which makes use of random forests and absolute position of points to compute a local metric; in [10], a local metric is learned based on a conical combination of Mahalanobis metrics and pair-wise similarities between the data; a last example of this non exhaustive list comes from [21], where the authors learn a mixture of local Mahalanobis distances.

3 Learning a perceptual color distance

In this section, we present a way to learn a perceptual distance that is invariant across acquisition conditions. First, we explain how we have created an image

[1] Note that kernel learning is another solution to consider non linearity in the data.

dataset designed for this purpose. Then, making use of the advantages of learning local metrics, we introduce our new algorithm that aims at accurately approximating a perceptual color distance in different parts of the RGB space. We end this section by a theoretical analysis of our algorithm.

3.1 Creating the dataset

Given two color patches, we want to design a perceptual distance not disturbed by the acquisition conditions. So we propose to use pairs of patches for which we can measure the true perceptual distance under standard viewing conditions and to image them under different other conditions.

The choice of the patches is key in this work since all the distances will be learned from these pairs. Consequently, the colors of the patches have to be well distributed in the RGB cube in order to be able to well approximate the color distance between two new pairs that have not been seen in the training set. Moreover, as we would like to learn a local perceptual distance, we need pairs of patches whose colors are close from each other. According to [22], ΔE_{00} seems to be a good candidate for that because it is designed to compare similar colors. Finally, since hue, chroma and luminance differences impact the perceptual color difference [22], the patches have to be chosen so that all these three variations are represented among the pairs.

Given these three requirements, we propose to use two different well-known sets of patches, namely the Farnsworth-Munsell 100 hue test and the Munsell atlas (see Fig. 1). The Farnsworth-Munsell 100 hue test is one of the most famous color vision tests which consists in ordering 84 patches in the correct order and any misplacement can point to some sort of color vision deficiency. Since these 84 patches are well distributed on the hue wheel, their colors will cover a large area of the RGB cube when imaging them under an important range of acquisition conditions. Furthermore, consecutive patches are known to have very small color differences and then, learning perceptual distances from such pairs is a good purpose. This set is constituting the main part of our dataset. Nevertheless, the colors of these patches first, are not highly saturated and second, they mostly exhibit hue variations and relatively small luminance and chroma differences. In order to cope with these weaknesses, we add to this dataset the 238 patches constituting the Munsell Student Color Set [19]. These patches are characterized by more saturated colors and the pairs of similar patches mostly exhibit luminance and chroma variations (since only the 5 principal and 5 intermediate hues are provided in this student set).

To build the dataset, we first use a spectroradiometer (Minolta CS 1000) in order to measure the spectra of each color patch of the Farnsworth set, the spectra of the Munsell atlas patches being available online [2]. Five measurements have been done in our light cabinet and the final spectra are the average of each measurement. From these spectra, we evaluate the $L^*a^*b^*$ coordinates of each patch under D65 illuminant. Then, we evaluate the distance ΔE_{00} between all

[2] https://www.uef.fi/spectral/spectral-database

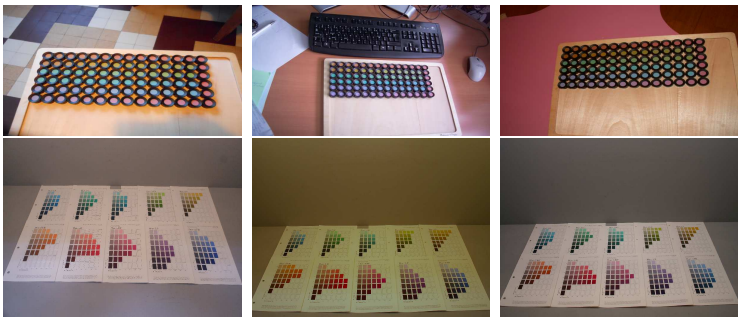


Fig. 1. Some images from our dataset showing (first row) the 84 used Farnsworth-Munsell patches or (second row) the 238 Munsell patches under different conditions.

the pairs of color patches [22]. Since we need patch pairs whose colors are similar, following the CIE recommendations (CIE Standard DS 014-6/E:2012), we select among the $C^2_{84} + C^2_{238}$ available pairs only the 223 that are characterized by a Euclidean distance in the CIELAB space (denoted ΔE_{ab}) less than 5.

Note that the available ΔE_{00} have been evaluated in the standard viewing conditions recommended by the CIE for color difference assessment and we would like to obtain these reference distances whatever the acquisition conditions. Consequently, we propose to use 4 different cameras, namely Kodak DCS Pro 14n, Konica Minolta Dimage Z3, Nikon Coolpix S6150 and Sony DCR-SR32 and a large variety of lights, viewpoints and backgrounds (since background also perturbs the colors of the patches). For each camera, we acquire 50 images of each Farnsworth pair and 15 of each Munsell pair (overall, 41,800 imaged pairs). Finally, after all these measurements and acquisitions, we have for each image of a pair, two image rendered RGB vectors and one reference distance ΔE_{00}.

3.2 Local metric learning algorithm

In this section, our objective is to approximate the reference distance ΔE_{00} by a metric learning approach in the RGB space which aims at optimizing K local metrics plus one global metric. For this task, we perform a preprocess by dividing the RGB space into K local parts thanks to a clustering step. From this, we deduce K+1 regions defining a partition $C_0, C_1, \ldots, C_K$ over the possible pairs of patches. A pair $p = (\mathbf{x}, \mathbf{x}')$ belongs to a region C_j, $1 \leq j \leq K$ if both $\mathbf{x}$ and $\mathbf{x}'$ belong to the same cluster j, otherwise p is assigned to region C_0. In other words, each region C_j corresponds to pairs related to cluster j, while C_0 contains the remaining pairs whose points do not belong to the same cluster. Then, we approximate ΔE_{00} by learning a Mahalanobis-like distance in every C_j $(j = 0, 1, \ldots, K)$, represented by its associated PSD 3×3 matrix $\mathbf{M_j}$.

Each metric learning step is done from a finite-size training sample of n_j triplets $T_j = \{(\mathbf{x_i}, \mathbf{x'_i}, \Delta E_{00})\}_{i=1}^{n_j}$ where $\mathbf{x_i}$ and $\mathbf{x'_i}$ represent color patches belonging to the same region C_j and $\Delta E_{00}(\mathbf{x_i}, \mathbf{x'_i})$ (ΔE_{00} for the sake of simplicity) their associated perceptual distance value. We define a loss function l on any pair of patches $(\mathbf{x}, \mathbf{x}')$: $l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) = \left| \Delta^2_{T_j} - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2 \right|$ where $\Delta_{T_j} =$

Algorithm 1: Local metric learning

input : A training set S of patches; a parameter $K \geq 2$
output: K local Mahalanobis distances and one global metric
begin
  Run K-means on S and deduce $K+1$ training subsets T_j $(j = 0, 1 \ldots, K)$ of triplets $T_j = \{(\mathbf{x_i}, \mathbf{x'_i}, \Delta E_{00})\}_{i=1}^{n_j}$ (where $\mathbf{x_i}, \mathbf{x'_i} \in C_j$ and $\Delta E_{ab}(\mathbf{x_i}, \mathbf{x'_i}) < 5$)
  for $j = 0 \rightarrow K$ **do**
    Learn $\mathbf{M_j}$ by solving the convex optimization Problem (1) using T_j

$\sqrt{(\mathbf{x} - \mathbf{x'})^T \mathbf{M_j} (\mathbf{x} - \mathbf{x'})}$, l measures the error made by a learned distance $\mathbf{M_j}$. We denote the empirical error over T_j by $\hat{\varepsilon}_{T_j}(\mathbf{M_j}) = \frac{1}{n_j} \sum_{(\mathbf{x}, \mathbf{x'}, \Delta E_{00}) \in T_j} l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x'}, \Delta E_{00}))$. We suggest to learn the matrix $\mathbf{M_j}$ that minimizes $\hat{\varepsilon}_{T_j}$ via the following regularized problem:

$$\underset{\mathbf{M_j} \succeq 0}{\arg\min} \ \hat{\varepsilon}_{T_j}(\mathbf{M_j}) + \lambda_j \|\mathbf{M_j}\|_{\mathcal{F}}^2, \tag{1}$$

where $\lambda_j > 0$ is a regularization parameter and $\| \cdot \|_{\mathcal{F}}$ denotes the Frobenius norm. To obtain a proper distance, $\mathbf{M_j}$ must be PSD (denoted by $\mathbf{M_j} \succeq 0$) and thus has to be projected onto the PSD cone as previously explained. Due to the simplicity of $\mathbf{M_j}$ (3×3 matrix), this operation is not costly [3]. It is worth noting that our optimization problem takes the form of a simple regularized least absolute deviation formulation. The interest of using the least absolute deviation, rather than a regularized least square, comes from the fact that it enables accurate estimates of small ΔE_{00} values.

The pseudo-code of our metric learning algorithm is presented in Alg. 1. Note that to solve the convex problem 1, we use a classical interior points approach. Moreover, parameter λ_j is tuned by cross-validation.

Discussion about Local versus Global Metric Note that in our approach, the metrics learned in the K regions $C_1, \ldots, C_K$ are local metrics while the one learned for region C_0 is rather a global metric considering pairs that do not fall in the same region. Beyond the fact that such a setting will allow us to derive generalization guarantees on our algorithm, it constitutes a straightforward solution to deal with patches at test time that would not be concerned by the same local metric in the color space. In this case, we make use of the matrix $\mathbf{M_0}$ associated to partition C_0. Another possible solution may consist in resorting to a Gaussian embedding of the local metrics. However, because this solution would imply learning additional parameters, we suggest in this paper to make use of this simple and efficient (parameters-wise) strategy. In the segmentation experiments of this paper, we will notice that $\mathbf{M_0}$ is used in only $\sim$20% of the cases. Finally, note that if $K = 1$, this boils down to learning only one global metric over the whole training sample. In the next section, we justify the consistency of this approach.

[3] We noticed during our experiments that $\mathbf{M_j}$ is, most of the time, PSD without requiring any projection on the cone.

3.3 Theoretical study

In this part, we provide a generalization bound justifying the consistency of our method. It is derived by considering (i) a multinomial distribution over the regions, and (ii) per region generalization guarantees that are obtained with the uniform stability framework [5].

We assume that the training sample $T = \cup_{j=0}^{K} T_j$ is drawn from an unknown distribution P such that for any $(\mathbf{x}, \mathbf{x}', \Delta E_{00}) \sim P$, $\Delta E_{00}(\mathbf{x}, \mathbf{x}') \leq \Delta_{\max}$, with $\Delta_{\max}$ the maximum distance value used in our context. We assume any input instance $\mathbf{x}$ to be normalized such that $\|\mathbf{x}\| \leq 1$, where $\|\cdot\|$ is the L2-norm[4].

The $K+1$ regions $C_0, \ldots, C_K$ define a partition of the support of P. In partition C_j, let $D_j = \max_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \sim P(C_j)}(\|\mathbf{x} - \mathbf{x}'\|)$ be the maximum distance between two elements and $P(C_j)$ be the marginal distribution.

Let $\mathbf{M} = \{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_K\}$ be the K+1 matrices learned by our Alg. 1. We define the true error associated to $\mathbf{M}$ by $\varepsilon(\mathbf{M}) = \sum_{j=0}^{K} \varepsilon_{P(C_j)}(\mathbf{M}_j) P(C_j)$ where $\varepsilon_{P(C_j)}(\mathbf{M}_j) = \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \sim P(C_j)} l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))$ is the local true risk for C_j. The empirical error over T of size n is defined as $\hat{\varepsilon}_T(\mathbf{M}) = \frac{1}{n} \sum_{j=0}^{K} n_j \hat{\varepsilon}_{T_j}(\mathbf{M_j})$ where $\hat{\varepsilon}_{T_j}(\mathbf{M_j}) = \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \in T_j} l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))$ is the empirical risk of T_j.

Generalization bound per region C_j To begin with, for any learned local matrix $\mathbf{M}_j$, we provide a bound on its associated local **true risk** $\varepsilon_{P(C_j)}(\mathbf{M}_j)$ in function of the **empirical risk** $\hat{\varepsilon}_{T_j}(\mathbf{M_j})$ over T_j.

Lemma 1 (Generalization bound per region). *With probability* $1 - \delta$*, for any matrix* $\mathbf{M}_j$ *related to a region* C_j*,* $0 \leq j \leq K$*, learned with Alg. 1, we have:*

$$|\varepsilon_{P(C_j)}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j})| \leq \frac{2D_j^4}{\lambda_j n_j} + \left(\frac{4D_j^4}{\lambda_j} + \Delta_{\max}\left(\frac{2D_j^2}{\sqrt{\lambda_j}} + 2\Delta_{\max}\right)\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n_j}}.$$

The proof of this lemma is provided in the supplementary material and is based on the uniform stability framework. It shows that the consistency is achieved in each region with a convergence rate in $O(1/\sqrt{n})$. When the region is compact, the quantity D_j is rather small making the bound tighter.

Generalization bound for Alg. 1 The generalization bound of our algorithm is based on the fact that the different marginals $P(C_j)$ can be interpreted as the parameters of a multinomial distribution. Thus, $(n_0, n_1, \ldots, n_K)$ is then a IID multinomial random variable with parameters $n = \sum_{j=0}^{n} n_j$ and $(P(C_0), P(C_1), \ldots, P(C_K))$. Our result makes use of the Bretagnolle-Huber-Carol concentration inequality for multinomial distributions [25] which is recalled in the supplementary material for the sake of completeness (this result has also been used in [35] in another context).

We are now ready to introduce the main theorem of the paper.

[4] Since we work in the RGB cube, any patch belongs to $[0; 255]^3$ and it is easy to normalize each coordinate by $255\sqrt{3}$.

Theorem 1 *Let $C_0, C_1, \ldots, C_k$ be the regions considered, then for any set of metrics $\mathbf{M} = \{\mathbf{M_0}, \ldots, \mathbf{M}_K\}$ learned by Alg. 1 from a data sample T of n pairs, we have with probability at least $1-\delta$ that*

$$\varepsilon(\mathbf{M}) \leq \hat{\varepsilon}_T(\mathbf{M}) + L_B\sqrt{\frac{2(K+1)\ln 2 + 2\ln 2/\delta}{n}} + \frac{2(KD^4+1)}{\lambda n}$$
$$+\left(\frac{4(KD^4+1)}{\lambda} + \Delta_{\max}\left(\frac{2(KD^2+1)}{\sqrt{\lambda}} + 2(K+1)\Delta_{\max}\right)\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n}},$$

where $D = \max_{1\leq j\leq K} D_j$, $L_B = \max\{\frac{\Delta_{max}}{\sqrt{\lambda}}, \Delta_{max}^2\}$ is the bound on the loss function and $\lambda = \min_{0\leq j\leq K} \lambda_j$ is the minimum regularization parameter among the $K+1$ learning problems used in Alg. 1.

The proof of this theorem is provided in the supplementary material. The first term after the empirical risk comes from the application of the Bretagnolle-Huber-Carol inequality with a confidence parameter $1-\delta/2$. The last terms are derived by applying the per region consistency Lemma 1 to all the regions with a confidence parameter $1-\delta/2(K+1)$ and the final result is derived thanks to the union bound.

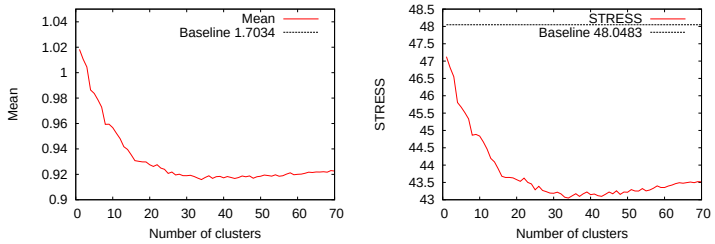
This result justifies the global consistency of our approach with a standard convergence rate in $O(1/\sqrt{n})$. We can remark that if the local regions $C_1, \ldots, C_n$ are rather small (*i.e.* D is significantly smaller than 1), then the last part of the bound will not suffer too much on the number of regions. On the other hand, there is also a trade-off between the number/size of regions considered and the number of instances falling in each region. It is important to have enough examples to learn good models.

4 Experiments

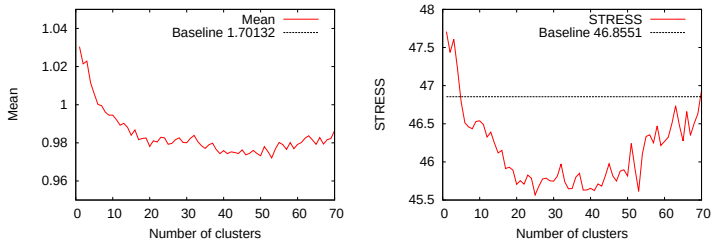
Evaluating the contribution of a metric learning algorithm can be done in two ways: (1) assessing the quality of the metric itself, and (2) measuring its impact once plugged in an application. In the following, we first evaluate the generalization ability of the learned metrics on our dataset. Then, we measure their contribution in a color segmentation application.

4.1 Evaluation on our dataset

To evaluate the generalization ability of the metrics, we conduct two experiments: We assess the behavior of our approach when it is applied (i) on new unseen colors and (ii) on new patches coming from a different unseen camera. In these experiments, we consider all the pairs of patches $(\mathbf{x}, \mathbf{x}')$ of our dataset characterized by a $\Delta E_{ab} < 5$, resulting in $41,800$ pairs. Due to the large amount of data, combined with the relative simplicity of the 3×3 local metrics, we notice that the algorithm is rather insensible to the choice of λ. Therefore, we use $\lambda = 1$ in all our experiments. The displayed results are the average over 5 runs.



(a) Generalization to new colors.



(b) Generalization to new cameras.

Fig. 2. (a): Generalization of the learned metrics to new colors; (b) Generalization of the learned metrics to new cameras. For (a) and (b), we plotted the Mean and STRESS values as a function of the number of clusters. The horizontal dashed line represents the STRESS baseline of $\widehat{\Delta E_{00}}$. For the sake of readability, **we have not plotted** the mean baseline of $\widehat{\Delta E_{00}}$ at 1.70.

To estimate the performance of our metric we use two criteria we want to make as small as possible. The first one is the mean absolute difference, computed over a test set TS, between the learned metric Δ_T - *i.e.* the metric learned with Alg. 1 - w.r.t. a training set of pairs T and the reference ΔE_{00}. As a second criterion, we use the STRESS[5] measure [17]. Roughly speaking, it evaluates quadratic differences between the learned metric Δ_T and the reference ΔE_{00}. We compare our approach to the state of the art where Δ_T is replaced by $\widehat{\Delta E_{00}}$ [22] in both criteria, i.e. transforming from rendered image RGB to $\widehat{L^*a^*b^*}$ and computing the $\widehat{\Delta E_{00}}$ distance.

Generalization to unseen colors In this experiment, we perform a 6-fold cross validation procedure over the set of *patches*. Thus we obtain, on average, 27927 training pairs and 13873 testing pairs. The results are shown on Fig. 2(a) according to an increasing number of clusters (from 1 to 70). We can see that using our learned metric Δ_T instead of the state of the art estimate $\widehat{\Delta E_{00}}$ [22] enables significant improvements according to both criteria (where the baselines are 1.70 for the mean and 48.05 for the STRESS). Note that from 50 clusters, the quality of the learned metric declines slightly while remaining much better than $\widehat{\Delta E_{00}}$. Figure 2(a) shows that $K = 20$ seems to be a good compromise between a high algorithmic complexity (the higher K, the larger the number of learned

[5] STandardized REsidual Sum of Squares.

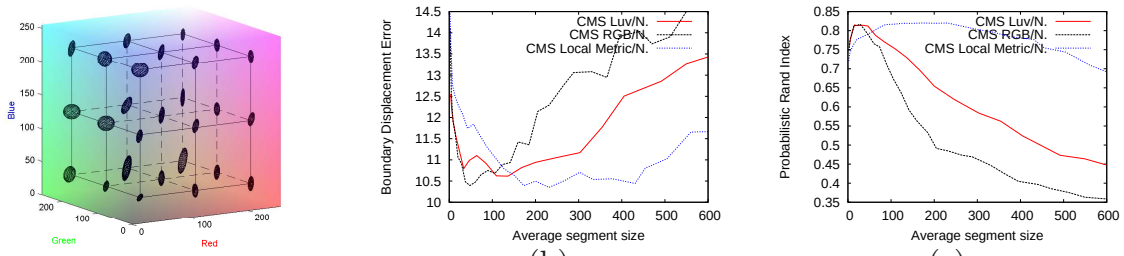
metrics) and good performances of the models. When $K = 20$, using a Student's t test over the mean absolute differences and a Fisher test over the STRESS, our method is significantly better than the state of the art with a p-value $< 1^{-10}$. Figure 2(a) also emphasizes the interest of learning several local metrics. Indeed, optimizing 20 local metrics rather than only one is significantly better with a p-value smaller than 0.001 for both criteria.

Generalization to unseen cameras In this experiment, our model is learned according to a 4-fold cross validation procedure such that each fold corresponds to the pairs coming from a given camera. Thus we learn the metric on a set of 31350 pairs and test it on a set of 10450 pairs. Therefore, this task is more complicated than before. The results are presented in Fig. 2(b). We can note that our approach always outperforms the state of the art for the mean criterion (of baseline 1.70). Regarding the STRESS, we are on average better when using between 5 to 60 clusters. Beyond 65 clusters, the performances decrease significantly. This behavior likely describes an overfitting phenomenon due to the fact that a lot of local metrics have been learned that are more and more specialized for 3 out of 4 cameras, and unable to generalize well to the fourth one. For this series of experiments, $K = 20$ is still a good value to deal with the trade-off between complexity and efficiency. Using a Student's t test over the mean absolute differences and a Fisher test over the STRESS, our method is significantly better with p-values respectively $< 1^{-10}$ and < 0.006. The interest of learning several local metrics rather than only one is still confirmed. Applying statistical comparison tests between $K = 20$ and $K = 1$ leads to small p-values < 0.001.

Thus for both series of experiments, $K = 20$ appears to be a good number of clusters and allows significant improvements. Therefore, we suggest to take this value in the next section to tackle a segmentation problem. Before that, let us finish this section by geometrically showing the interest of learning local metrics. Figure 3(a) shows ellipsoids uniformly distributed in the RGB space whose surface corresponds to the RGB colors lying at the corresponding learned local distance of 1 from the center of the ellipsoid. It is worth noting that the variability of the shapes and orientations of the ellipsoids is high, meaning that each local metric could capture local specificities of the color space. The experimental results presented in the next section will prove this claim.

4.2 Application to image segmentation

In this experiment, we evaluate the performance of our approach in a color based image segmentation application. We propose to use the approach from [4] that suggests a nice extension of the classical mean-shift algorithm by accounting color information. Furthermore, the authors show that the more perceptual the used distance, the better the results. Especially, by using the default transform from the available camera RGB to the $\widehat{L^*u^*v^*}$, they significantly improve the segmentation results over the simple RGB coordinates. Our aim is not to propose a new segmentation algorithm but to use the exact algorithm proposed



(a) (b) (c)

Fig. 3. (a) Interest of learning local metrics. We took 27 points uniformly distributed on the RGB cube. Around each point we plotted an ellipsoid where the surface corresponds to the RGB colors lying at a learned distance of 1. In this case we used the metric learned by our algorithm using $K = 20$. (b) Boundary Displacement Error (lower is better) versus the average segment size. (c) Probabilistic Rand Index (higher is better) versus the average segment size.

in [4] working in the RGB space and to replace in their code (publicly available) the distance between two colors with our learned color distance Δ_T. By this way, we can compare the perceptual property of our distance with this of the recommended default approach (euclidean distance in the $\widehat{L^*u^*v^*}$ space).

Therefore, we take exactly the same protocol as [4]. We use the same 200 images taken from the well-known Berkeley dataset and the associated ground-truth that is constituted by 1087 segmented images provided by humans. In order to assess the quality of the segmentation, as recommended by [4], we use the average Boundary Displacement Error (BDE) and the Probabilistic Rand Index (PRI). Note that the better the quality of the segmentation, the lower the BDE and the higher the PRI. The segmentation algorithm proposed in [4] has one main parameter which is the color distance threshold under which two neighbor pixels (or sets of pixels) have to be merged in the same segment. As in [4], we plot the evolution of the quality criteria versus the average segment size (see Figs. 3(b) and 3(c)). For comparison, we have run the code from [4] for the parameters providing the best results in their paper, namely "CMS Luv/N.", corresponding to their color mean-shift (CMS) applied in the $\widehat{L^*u^*v^*}$ color space. The results of CMS applied in the RGB color space with the classical euclidean distance are plotted as "CMS RGB/N." and those of CMS applied with our color distance in the RGB color space are plotted as "CMS Local Metric/N.".

For both criteria, we can see that our learned color distance significantly improves the quality of the results over the two other approaches, i.e. it provides a segmentation that is closer to the one computed by humans. This is truer when the segment size is increasing (right part of the plots). It is important to understand that increasing the average segment size (moving to the right on the plots) is like merging neighbor segments in the images. So by analyzing the curves, we can see that for the classical approaches ("CMS Luv/N." and "CMS RGB/N."), it seems that the segments that are merged together when moving

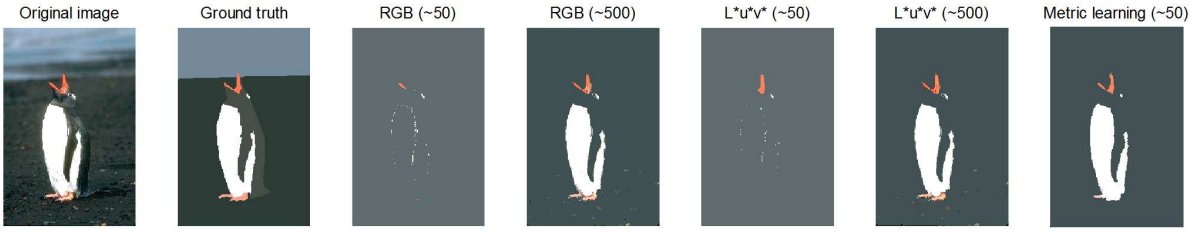


Fig. 4. Segmentation illustration. When the number of clusters is low (around 50), the segmentation provided by RGB or $\widehat{L^*u^*v^*}$ are far from the ground truth, unlike our approach which provides nice results. To get the same perceptual result, both methods require about 500 clusters.

to the right on the plot are not the ones that would be merged by humans. That is why both criteria are worst (BDE increases and PRI decreases) on the right for these methods. On the other hand, it seems that our distance is more accurate when merging neighbor segments since for high average segment sizes, our results are much better. This point can be observed in Fig. 4, where the segment size is high, i.e. when the number of clusters is low (50), the segmentation provided by RGB or $\widehat{L^*u^*v^*}$ are far from the ground truth, unlike our approach which provides nice results. To get the same perceptual result, both methods require about 500 clusters. We provide more segmentation comparisons in the supplementary material.

5 Conclusion

In this paper, we presented a new local metric learning approach for approximating perceptual distances directly in the rendered image RGB space. Our method outperforms the state of the art for generalizing to unseen colors and to unseen camera distortions and also in a color image segmentation task. The model is both efficient - for each pair one only needs to find the two clusters of the patches and to apply a 3×3 matrix - and expressive thanks to the local aspect allowing us to model different distortions in the RGB space. Moreover, we derived a generalization bound ensuring the consistency of the learning approach. Finally, we designed a dataset of color patches which can play the role of a benchmark for the computer vision community.

Future work will include the use of metric combination approaches together with more complex regularizers on the set of models (mixed and nuclear norms for example). Another perspective concerns the spatial continuity of the learned metrics. Even though Fig. 3(a) shows ellipsoids that tend to be locally regular leading to a certain spatial continuity, our model does not explicitely deal with this issue. One solution may consist in resorting to a Gaussian embedding of the local metrics. From a practical side, the development of transfer learning methods for improving the generalization to unknown devices could be an interesting direction. Another different perspective would be to learn photometric invariant distances.

Acknowledgments

The final publication is available at `http://link.springer.com/`.

References

1. Achanta, R., Susstrunk, S.: Saliency detection using maximum symmetric surround. In: Proc. of ICIP. pp. 2653–2656. Hong Kong (2010)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik., J.: Contour detection and hierarchical image segmentation. IEEE Trans. on PAMI 33(5), 898–916 (2011)
3. Bellet, A., Habrard, A., Sebban, M.: A survey on metric learning for feature vectors and structured data (arxiv:1306.6709v3). Tech. rep. (August 2013)
4. Bitsakos, K., Fermüller, C., Aloimonos, Y.: An experimental study of color-based segmentation algorithms based on the mean-shift concept. In: Proc. of ECCV. pp. 506–519. Greece (2010)
5. Bousquet, O., Elisseeff, A.: Stability and generalization. JMLR 2, 499–526 (2002)
6. Burghouts, G., Geusebroek, J.M.: Performance evaluation of local colour invariants. Computer Vision and Image Understanding 113 (1), 48–62 (2009)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of CVPR. pp. 886–893 (2005)
8. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proc. of ICML. pp. 209–216 (2007)
9. Huang, M., Liu, H., Cui, G., Luo, M.R., Melgosa, M.: Evaluation of threshold color differences using printed samples. JOSA A 29(6), 883–891 (2012)
10. Huang, Y., Li, C., Georgiopoulos, M., Anagnostopoulos, G.C.: Reduced-rank local distance metric learning. In: Proc. of ECML/PKDD (3). pp. 224–239 (2013)
11. Khan, R., van de Weijer, J., Khan, F., Muselet, D., Ducottet, C., Barat, C.: Discriminative color descriptor. In: Proc. of CVPR. Portland, USA (2013)
12. Kim, S.J., Lin, H.T., Lu, Z., Süsstrunk, S., Lin, S., Brown, M.S.: A new in-camera imaging model for color computer vision and its application. IEEE Trans. Pattern Anal. Mach. Intell. 34(12), 2289–2302 (2012)
13. Kim, S., Lin, H., Lu, Z., Susstrunk, S., Lin, S., Brown, M.S.: A new in-camera imaging model for color computer vision and its application. IEEE Trans. on PAMI 34(12), 2289–2302 (2012)
14. Larraín, R., Schaefer, D., Reed, J.: Use of digital images to estimate {CIE} color coordinates of beef. Food Research Int. 41(4), 380 – 385 (2008)
15. Len, K., Mery, D., Pedreschi, F., Len, J.: Color measurement in l*a*b* units from rgb digital images. Food Research Int. 39(10), 1084 – 1091 (2006)
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
17. Melgosa, M., Huertas, R., Berns, R.: Performance of recent advanced color-difference formulas using the standardized residual sum of squares index. JOSA A 25(7), 1828–34 (2008)
18. Mojsilovic, A.: A computational model for color naming and describing color composition of images. IEEE Trans. on Image Processing 14(5), 690–699 (May 2005)
19. Munsell, A.H.: A pigment color system and notation. The American Journal of Psychology 23(2), 236–244 (1912)
20. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Trans. on PAMI 32(9), 1582–1596 (2010)

21. Semerci, M., Alpaydin, E.: Mixtures of large margin nearest neighbor classifiers. In: Proc. of ECML/PKDD (2). pp. 675–688 (2013)
22. Sharma, G., Wu, W., Dalal, E.: The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. Color Research Applications 30, 21–30 (2005)
23. Stokes, M., Anderson, M., Chandrasekar, S., Motta, R.: A standard default color space for the internet: sRGB. Tech. rep., Hewlett-Packard and Microsoft (1996), http://www.w3.org/Graphics/Color/sRGB.html
24. Union, I.T.: Parameter values for the hdtv standards for production and international programme exchange, itu-r recommendation bt.709-4. Tech. rep. (March 2000)
25. van der Vaart, A.W., Wellner, J.A.: Weak convergence and empirical processes. Springer (2000)
26. Wang, J., Kalousis, A., Woznica, A.: Parametric local metric learning for nearest neighbor classification. In: Proc. of NIPS. pp. 1610–1618 (2012)
27. J. van de Weijer, T.G., Bagdanov, A.: Boosting color saliency in image feature detection. IEEE Trans. on PAMI 28(1), 150–156 (2006)
28. J. van de Weijer, T.G., Geusebroek, J.: Edge and corner detection by photometric quasi-invariants. IEEE Trans. on PAMI 27(4), 1520–1526 (2005)
29. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: Proc. of NIPS (2006)
30. Weinberger, K., Saul, L.: Distance metric learning for large margin nearest neighbor classification. JMLR 10, 207–244 (2009)
31. Wyszecki, G., Stiles, W.S.: Color Science: Concepts and Methods, Quantitative Data and Formulas. John Wiley & Sons Inc, 2nd revised ed., New York (2000)
32. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Proc. NIPS. pp. 505–512 (2002)
33. Xiong, C., Johnson, D., Xu, R., Corso, J.J.: Random forests for metric learning with implicit pairwise position dependence. In: Proc. of KDD. pp. 958–966. ACM (2012)
34. Xiong, Y., Saenko, K., Darrell, T., Zickler, T.: From pixels to physics: Probabilistic color de-rendering. In: Proc. of CVPR. Providence, USA (2012)
35. Xu, H., Mannor, S.: Robustness and Generalization. Machine Learning 86(3), 391–423 (2012)
36. Xue, W., Mou, X., Zhang, L., Feng, X.: Perceptual fidelity aware mean squared error. In: Proc. of ICCV (2013)
37. Freely avaible on the authors' personal web pages.

Modeling Perceptual Color Differences by Local Metric Learning

Supplementary Material 1

Michaël Perrot, Amaury Habrard, Damien Muselet, and Marc Sebban

LaHC, UMR CNRS 5516, Université Jean-Monnet, F-42000, Saint-Étienne, France
{michael.perrot,amaury.habrard,damien.muselet,marc.sebban}@univ-st-etienne.fr

1 Overview of the supplementary material

This supplementary material is organised into two parts. In Section 2 we provide the proofs of the lemma and the theorem presented in Section 3.3 of the paper, while Section 3 presents some examples of image segmentation.

2 Theoretical analysis

This section presents the proofs of Lemma 1 and Theorem 1 from Section 3.3 of the paper. Lemma 1 is proved in Section 2.1 and Theorem 1 is proved in Section 2.2.

2.1 Generalization bound per region C_j

First, we recall our optimization problem considered in each region C_j:

$$\arg\min_{\mathbf{M_j} \succeq 0} F_{T_j}(\mathbf{M_j}) \tag{1}$$

where

$$\begin{aligned} F_{T_j}(\mathbf{M_j}) &= \hat{\varepsilon}_{T_j}(\mathbf{M_j}) + \lambda_j \|\mathbf{M_j}\|_{\mathcal{F}}^2, \\ \hat{\varepsilon}_{T_j}(\mathbf{M_j}) &= \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \in T_j} l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})), \\ \text{and } \ l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) &= \left| (\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') - \Delta E_{00}\left(\mathbf{x}, \mathbf{x}'\right)^2 \right|. \end{aligned}$$

Here $\hat{\varepsilon}_{T_j}(\mathbf{M_j})$ stands for the empirical risk of a matrix $\mathbf{M_j}$ over a training set T_j, of size n_j, drawn from an unknown distribution $P(C_j)$. The true risk $\varepsilon_{P(C_j)}(\mathbf{M_j})$ is defined as follows:

$$\varepsilon_{P(C_j)}(\mathbf{M_j}) = \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \sim P(C_j)} \left[l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) \right].$$

In this section, T_j^i denotes the training set obtained from T_j by replacing the i^{th} example of T_j by a new independent one. Moreover, we have $\Delta_{\max} = \max_{0 \leq j \leq K} \left\{ \max_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \sim P(C_j)} \left\{ \Delta E_{00}\left(\mathbf{x}, \mathbf{x}'\right) \right\} \right\}$ and $D_j = \max_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \sim P(C_j)} \left(\|\mathbf{x} - \mathbf{x}'\| \right) \leq 1$[1].

To derive such a generalization bound, we need to consider loss functions that fulfill two properties: k-lipschitz continuity (Definition A) and (σ, m)-admissibility (Definition B).

[1] We assume the examples to be normalized such that $\|\mathbf{x}\| \leq 1$.

Definition A (k-lipschitz continuity) *A loss function $l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))$ is k-lipschitz w.r.t. its first argument if, for any matrices $\mathbf{M_j}$, $\mathbf{M'_j}$ and any example $(\mathbf{x}, \mathbf{x}', \Delta E_{00})$, there exists $k \geq 0$ such that:*

$$\left|l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) - l(\mathbf{M'_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))\right| \leq k\|\mathbf{M_j} - \mathbf{M'_j}\|_{\mathcal{F}}.$$

This k-lipschitz property ensures that the loss deviation does not exceed the deviation between matrices $\mathbf{M_j}$ and $\mathbf{M'_j}$ with respect to a positive constant k.

Definition B ((σ, m)-admissibility) *A loss function $l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))$ is (σ, m)-admissible, w.r.t. $\mathbf{M_j}$, if it is convex w.r.t. its first argument and for two examples $(\mathbf{x}, \mathbf{x}', \Delta E_{00}(\mathbf{x}, \mathbf{x}'))$ and $(\mathbf{t}, \mathbf{t}', \Delta E_{00}(\mathbf{t}, \mathbf{t}'))$, we have:*

$$\left|l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}(\mathbf{x}, \mathbf{x}'))) - l(\mathbf{M_j}, (\mathbf{t}, \mathbf{t}', \Delta E_{00}(\mathbf{t}, \mathbf{t}')))\right| \leq \sigma \left|\Delta E_{00}(\mathbf{x}, \mathbf{x}') - \Delta E_{00}(\mathbf{t}, \mathbf{t}')\right| + m.$$

Definition B bounds the difference between the losses of two examples by a value only related to the ΔE_{00} values plus a constant independent from $\mathbf{M_j}$. Let us introduce a last concept which is required to derive a generalization bound.

Definition C (Uniform stability) *In a region C_j, a learning algorithm has a uniform stability in $\frac{\mathcal{K}}{n_j}$, with $\mathcal{K} \geq 0$ a constant, if $\forall i$,*

$$\sup_{(\mathbf{x}, \mathbf{x}', \Delta E_{00}) \sim P(C_j)} \left|l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) - l(\mathbf{M^i_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))\right| \leq \frac{\mathcal{K}}{n_j},$$

where $\mathbf{M_j}$ is the matrix learned on the training set T_j and $\mathbf{M^i_j}$ is the matrix learned on the training set T^i_j.

The uniform stability guarantees that the solutions learned with two close training sets are not significantly different and that the variation converges in $O(1/n_j)$.

To prove Lemma 1 of the paper, we need several additional lemmas and one more theorem which are not presented in the paper. First we show that our loss is k-lipschitz continuous, (σ, m)-admissible and that our algorithm respects the property of uniform stability. For the sake of readability, we number these lemmas and this theorem with capital letters.

Lemma A (k-lipschitz continuity) *Let $\mathbf{M_j}$ and $\mathbf{M'_j}$ be two matrices for a region C_j and $(\mathbf{x}, \mathbf{x}', \Delta E_{00})$ be an example. Our loss $l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))$ is k-lipschitz with $k = D_j^2$.*

Proof.

$$\begin{aligned}
&\left|l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) - l(\mathbf{M'_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))\right| \\
&\quad = \left|\left|(\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2\right| - \left|(\mathbf{x} - \mathbf{x}')^T \mathbf{M'_j} (\mathbf{x} - \mathbf{x}') - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2\right|\right| \\
&\quad \leq \left|(\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') - (\mathbf{x} - \mathbf{x}')^T \mathbf{M'_j} (\mathbf{x} - \mathbf{x}')\right| \qquad (2.1) \\
&\quad = \left|(\mathbf{x} - \mathbf{x}')^T \left(\mathbf{M_j} - \mathbf{M'_j}\right) (\mathbf{x} - \mathbf{x}')\right| \\
&\quad \leq \|\mathbf{x} - \mathbf{x}'\| \|\mathbf{M_j} - \mathbf{M'_j}\|_{\mathcal{F}} \|\mathbf{x} - \mathbf{x}'\| \qquad (2.2) \\
&\quad \leq D_j^2 \|\mathbf{M_j} - \mathbf{M'_j}\|_{\mathcal{F}} \qquad (2.3)
\end{aligned}$$

Inequality (2.1) is due to the triangle inequality, (2.2) is obtained by application of the Cauchy-Schwarz inequality and some classical norm properties. (2.3) comes from the definition of D_j. Setting $k = D_j^2$ gives the Lemma.

We now provide a lemma that will help to prove Lemma C on the (σ, m)-admissibility of our loss function.

Lemma B *Let $\mathbf{M_j}$ be an optimal solution of Problem* (1)*, we have*

$$\|\mathbf{M_j}\| \leq \frac{\Delta_{\max}}{\sqrt{\lambda_j}}.$$

Proof. Since $\mathbf{M_j}$ is an optimal solution of Problem (1), we have then:

$$
\begin{aligned}
& F_{T_j}(\mathbf{M_j}) \leq F_{T_j}(\mathbf{0}) \\
\Leftrightarrow \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \in T_j} & l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) + \lambda_j \|\mathbf{M_j}\|_{\mathcal{F}}^2 \leq \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \in T_j} l(\mathbf{0}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) + \lambda_j \|\mathbf{0}\|_{\mathcal{F}}^2 \\
\Rightarrow & \quad \lambda_j \|\mathbf{M_j}\|_{\mathcal{F}}^2 \leq \frac{1}{n_j} \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00}) \in T_j} l(\mathbf{0}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) \quad (3.1) \\
\Rightarrow & \quad \lambda_j \|\mathbf{M_j}\|_{\mathcal{F}}^2 \leq \Delta_{\max}^2 \quad (3.2) \\
\Rightarrow & \quad \|\mathbf{M_j}\|_{\mathcal{F}} \leq \frac{\Delta_{\max}}{\sqrt{\lambda_j}}.
\end{aligned}
$$

Inequality (3.1) comes from the fact that our loss is always positive and that $\|\mathbf{0}\|_{\mathcal{F}} = 0$. (3.2) is obtained by noting that $l(\mathbf{0}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) \leq \Delta_{\max}^2$.

Lemma C ((σ, m)-admissibility) *Let $(\mathbf{x}, \mathbf{x}', \Delta E_{00}(\mathbf{x}, \mathbf{x}'))$ and $(\mathbf{t}, \mathbf{t}', \Delta E_{00}(\mathbf{t}, \mathbf{t}'))$ be two examples and $\mathbf{M_j}$ be the optimal solution of Problem* (1)*. The loss $l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))$ is (σ, m)-admissible with $\sigma = 2\Delta_{\max}$ and $m = \frac{2D_j^2 \Delta_{\max}}{\sqrt{\lambda_j}}$.*

Proof.

$$
\begin{aligned}
& \left| l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}(\mathbf{x}, \mathbf{x}'))) - l(\mathbf{M_j}, (\mathbf{t}, \mathbf{t}', \Delta E_{00}(\mathbf{t}, \mathbf{t}'))) \right| \\
& \quad = \left| \left| (\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2 \right| - \left| (\mathbf{t} - \mathbf{t}')^T \mathbf{M_j} (\mathbf{t} - \mathbf{t}') - \Delta E_{00}(\mathbf{t}, \mathbf{t}')^2 \right| \right| \\
& \quad \leq \left| (\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') - (\mathbf{t} - \mathbf{t}')^T \mathbf{M_j} (\mathbf{t} - \mathbf{t}') \right| + \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}')^2 - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2 \right| \quad (4.1) \\
& \quad \leq \left| (\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') \right| + \left| (\mathbf{t} - \mathbf{t}')^T \mathbf{M_j} (\mathbf{t} - \mathbf{t}') \right| + \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}')^2 - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2 \right| \quad (4.2) \\
& \quad \leq 2 \max_{(\mathbf{x},\mathbf{x}')} \left\{ \left| (\mathbf{x} - \mathbf{x}')^T \mathbf{M_j} (\mathbf{x} - \mathbf{x}') \right| \right\} + \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}')^2 - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2 \right| \\
& \quad \leq \frac{2D_j^2 \Delta_{\max}}{\sqrt{\lambda_j}} + \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}')^2 - \Delta E_{00}(\mathbf{x}, \mathbf{x}')^2 \right| \quad (4.3) \\
& \quad \leq \frac{2D_j^2 \Delta_{\max}}{\sqrt{\lambda_j}} + \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}') + \Delta E_{00}(\mathbf{x}, \mathbf{x}') \right| \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}') - \Delta E_{00}(\mathbf{x}, \mathbf{x}') \right| \\
& \quad \leq \frac{2D_j^2 \Delta_{\max}}{\sqrt{\lambda_j}} + 2\Delta_{\max} \left| \Delta E_{00}(\mathbf{t}, \mathbf{t}') - \Delta E_{00}(\mathbf{x}, \mathbf{x}') \right|.
\end{aligned}
$$

Inequalities (4.1) and (4.2) are obtained by applying the triangle inequality respectively twice and once, (4.3) comes from the fact that $\|\mathbf{M_j}\|_{\mathcal{F}} \leq \frac{\Delta_{\max}}{\sqrt{\lambda_j}}$ (Lemma B) and that $\|\mathbf{x} - \mathbf{x}'\| \leq D_j$. Setting $\sigma = 2\Delta_{\max}$ and $m = \frac{2D_j^2 \Delta_{\max}}{\sqrt{\lambda_j}}$ gives the Lemma.

We will now prove the uniform stability of our algorithm but before to present this proof, we need the following Lemma.

Lemma D *Let $F_{T_j}()$ and $F_{T_j^i}()$ be the functions to optimize, $\mathbf{M_j}$ and $\mathbf{M_j^i}$ their corresponding minimizers, and λ_j the regularization parameter used in our algorithm. Let $\Delta\mathbf{M_j} = \mathbf{M_j} - \mathbf{M_j^i}$, then, we have, for any $t \in [0, 1]$,*

$$
\|\mathbf{M_j}\|_{\mathcal{F}}^2 - \|\mathbf{M_j} - t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 + \|\mathbf{M_j^i}\|_{\mathcal{F}}^2 - \|\mathbf{M_j^i} + t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 \leq \frac{2kt}{\lambda_j n_j} \|\Delta\mathbf{M_j}\|_{\mathcal{F}}. \quad (5)
$$

Proof. This proof is similar to the proof of Lemma 20 in [1] which we recall for the sake of completeness. $\hat{\varepsilon}_{T_j^i}()$ is a convex function, thus, for any $t \in [0,1]$, we can write:

$$\hat{\varepsilon}_{T_j^i}(\mathbf{M_j} - t\Delta\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j}) \leq t\left(\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j})\right), \tag{6}$$

$$\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i} + t\Delta\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) \leq t\left(\hat{\varepsilon}_{T_j^i}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i})\right). \tag{7}$$

By summing inequalities (6) and (7) we obtain

$$\hat{\varepsilon}_{T_j^i}(\mathbf{M_j} - t\Delta\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j}) + \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i} + t\Delta\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) \leq 0. \tag{8}$$

Since $\mathbf{M_j}$ and $\mathbf{M_j^i}$ are minimizers of $F_{T_j}()$ and $F_{T_j^i}()$, we can write:

$$F_{T_j}(\mathbf{M_j}) - F_{T_j}(\mathbf{M_j} - t\Delta\mathbf{M_j}) \leq 0, \tag{9}$$

$$F_{T_j^i}(\mathbf{M_j^i}) - F_{T_j^i}(\mathbf{M_j^i} + t\Delta\mathbf{M_j}) \leq 0. \tag{10}$$

By summing inequalities (9) and (10), we obtain

$$\begin{aligned}\hat{\varepsilon}_{T_j}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j} - t\Delta\mathbf{M_j}) + \lambda_j\|\mathbf{M_j}\|_{\mathcal{F}}^2 - \lambda_j\|\mathbf{M_j} - t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 + \\ \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i} + t\Delta\mathbf{M_j}) + \lambda_j\|\mathbf{M_j^i}\|_{\mathcal{F}}^2 - \lambda_j\|\mathbf{M_j^i} + t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 \leq 0.\end{aligned} \tag{11}$$

We can now sum inequalities (8) and (11) to obtain

$$\begin{aligned}\hat{\varepsilon}_{T_j}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j} - t\Delta\mathbf{M_j}) + \hat{\varepsilon}_{T_j^i}(\mathbf{M_j} - t\Delta\mathbf{M_j}) + \\ \lambda_j\|\mathbf{M_j}\|_{\mathcal{F}}^2 - \lambda_j\|\mathbf{M_j} - t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 + \lambda_j\|\mathbf{M_j^i}\|_{\mathcal{F}}^2 - \lambda_j\|\mathbf{M_j^i} + t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 \leq 0.\end{aligned} \tag{12}$$

From (12), we can write:

$$\lambda_j\|\mathbf{M_j}\|_{\mathcal{F}}^2 - \lambda_j\|\mathbf{M_j} - t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 + \lambda_j\|\mathbf{M_j^i}\|_{\mathcal{F}}^2 - \lambda_j\|\mathbf{M_j^i} + t\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 \leq B \tag{13}$$

with

$$B = \hat{\varepsilon}_{T_j^i}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j}) + \hat{\varepsilon}_{T_j}(\mathbf{M_j} - t\Delta\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j} - t\Delta\mathbf{M_j}).$$

We are now looking for a bound on B:

$$\begin{aligned}
B &\leq \left|\hat{\varepsilon}_{T_j}(\mathbf{M_j} - t\Delta\mathbf{M_j}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j} - t\Delta\mathbf{M_j}) + \hat{\varepsilon}_{T_j^i}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right| \\
&\leq \frac{1}{n_j}\left|\sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T_j} l(\mathbf{M_j} - t\Delta\mathbf{M_j}, (\mathbf{x},\mathbf{x}',\Delta E_{00})) - \sum_{(\mathbf{t},\mathbf{t}',\Delta E_{00})\in T_j^i} l(\mathbf{M_j} - t\Delta\mathbf{M_j}, (\mathbf{t},\mathbf{t}',\Delta E_{00})) + \right. \\
&\qquad \left. \sum_{(\mathbf{t},\mathbf{t}',\Delta E_{00})\in T_j^i} l(\mathbf{M_j}, (\mathbf{t},\mathbf{t}',\Delta E_{00})) - \sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T_j} l(\mathbf{M_j}, (\mathbf{x},\mathbf{x}',\Delta E_{00}))\right| \\
&= \frac{1}{n_j}\left|l(\mathbf{M_j} - t\Delta\mathbf{M_j}, (\mathbf{x_i},\mathbf{x_i}',\Delta E_{00})) - l(\mathbf{M_j} - t\Delta\mathbf{M_j}, (\mathbf{t_i},\mathbf{t_i}',\Delta E_{00})) + \right. \\
&\qquad \left. l(\mathbf{M_j}, (\mathbf{t_i},\mathbf{t_i}',\Delta E_{00})) - l(\mathbf{M_j}, (\mathbf{x_i},\mathbf{x_i}',\Delta E_{00}))\right| \qquad (14.1) \\
&\leq \frac{1}{n_j}\left(\left|l(\mathbf{M_j} - t\Delta\mathbf{M_j}, (\mathbf{x_i},\mathbf{x_i}',\Delta E_{00})) - l(\mathbf{M_j}, (\mathbf{x_i},\mathbf{x_i}',\Delta E_{00}))\right| + \right. \\
&\qquad \left.\left|l(\mathbf{M_j}, (\mathbf{t_i},\mathbf{t_i}',\Delta E_{00})) - l(\mathbf{M_j} - t\Delta\mathbf{M_j}, (\mathbf{t_i},\mathbf{t_i}',\Delta E_{00}))\right|\right) \qquad (14.2) \\
&\leq \frac{1}{n_j}\left(k\|\mathbf{M_j} - t\Delta\mathbf{M_j} - \mathbf{M_j}\|_{\mathcal{F}} + k\|\mathbf{M_j} - \mathbf{M_j} + t\Delta\mathbf{M_j}\|_{\mathcal{F}}\right) \qquad (14.3) \\
&\leq \frac{2kt}{n_j}\|\Delta\mathbf{M_j}\|_{\mathcal{F}}.
\end{aligned}$$

Equality (14.1) comes from the fact that T_j and T_j^i only differ by their i^{th} example, inequality (14.2) is due to the triangle inequality and (14.3) is obtained thanks to the k-lipschitz property of our loss (Lemma A).

Then combining the bound on B with equation (13) and dividing both sides by λ_j gives the Lemma.

We can now show the uniform stability property of the approach.

Lemma E (Uniform stability) *Given a training sample T_j of n_j examples drawn i.i.d. from $P(C_j)$, our algorithm has a uniform stability in $\frac{\mathcal{K}}{n_j}$ with $\mathcal{K} = \frac{2D_j^4}{\lambda_j}$.*

Proof. By setting $t = \frac{1}{2}$ in Lemma D, one can obtain for the left hand side:

$$\|\mathbf{M_j}\|_{\mathcal{F}}^2 - \|\mathbf{M_j} - \frac{1}{2}\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 + \|\mathbf{M_j^i}\|_{\mathcal{F}}^2 - \|\mathbf{M_j^i} + \frac{1}{2}\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 = \frac{1}{2}\|\Delta\mathbf{M_j}\|_{\mathcal{F}}^2$$

and thus:

$$\frac{1}{2}\|\Delta\mathbf{M_j}\|_{\mathcal{F}}^2 \leq \frac{2k\frac{1}{2}}{\lambda_j n_j}\|\Delta\mathbf{M_j}\|_{\mathcal{F}},$$

which implies

$$\|\Delta\mathbf{M_j}\|_{\mathcal{F}} \leq \frac{2k}{\lambda_j n_j}.$$

Since our loss is k-lipschitz (Lemma A) we have:

$$\begin{aligned}\left|l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) - l(\mathbf{M_j^i}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))\right| &\leq k\|\Delta\mathbf{M_j}\|_{\mathcal{F}} \\ &\leq \frac{2k^2}{\lambda_j n_j}.\end{aligned}$$

In particular,

$$\sup_{(\mathbf{x},\mathbf{x}',\Delta E_{00})} \left|l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) - l(\mathbf{M'_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00}))\right| \leq \frac{2k^2}{\lambda_j n_j}.$$

By recalling that $k = D_j^2$ (Lemma A) and setting $\mathcal{K} = \frac{2k^2}{\lambda_j}$, we get the lemma.

We now recall the McDiarmid inequality [2], used to prove our main theorem.

Theorem A (McDiarmid inequality) *Let $X_1, ..., X_n$ be n independent random variables taking values in X and let $Z = f(X_1, ..., X_n)$. If for each $1 \leq i \leq n$, there exists a constant c_i such that*

$$\sup_{x_1,...,x_n,x_i' \in \mathcal{X}} |f(x_1, ..., x_n) - f(x_1, ..., x_i', ..., x_n)| \leq c_i, \forall 1 \leq i \leq n,$$

$$\textit{then for any } \epsilon > 0, \textit{Pr}\left[|Z - \mathbb{E}[Z]| \geq \epsilon\right] \leq 2\exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

Using Lemma E about the stability of our algorithm and the McDiarmid inequality we can derive our generalization bound. For this purpose, we replace Z by $R_{T_j} = \varepsilon_{P(C_j)}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j})$ in Theorem A and we need to bound $\mathbb{E}_{T_j}\left[R_{T_j}\right]$ and $\left|R_{T_j} - R_{T_j^i}\right|$, which is done in the following two lemmas.

Lemma F *For any learning method of estimation error R_{T_j} and satisfying a uniform stability in $\frac{\mathcal{K}}{n_j}$, we have*

$$\mathbb{E}_{T_j}\left[R_{T_j}\right] \leq \frac{\mathcal{K}}{n_j}.$$

Proof.

$$\begin{aligned}
\mathbb{E}_{T_j}\left[R_{T_j}\right] &\leq \mathbb{E}_{T_j}\left[\mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right] \\
&\leq \mathbb{E}_{T_j,(\mathbf{x},\mathbf{x}',\Delta E_{00})}\left[\left| l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00})) - \frac{1}{n_j}\sum_{(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00})\in T_j} l(\mathbf{M_j},(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00}))\right|\right] \\
&\leq \mathbb{E}_{T_j,(\mathbf{x},\mathbf{x}',\Delta E_{00})}\left[\left| \frac{1}{n_j}\sum_{(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00})\in T_j} \left(l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00})) - l(\mathbf{M_j},(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00}))\right)\right|\right] \\
&\leq \mathbb{E}_{T_j,(\mathbf{x},\mathbf{x}',\Delta E_{00})}\left[\left| \frac{1}{n_j}\sum_{(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00})\in T_j} \left(l(\mathbf{M_j^k},(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00})) - l(\mathbf{M_j},(\mathbf{x_k},\mathbf{x'_k},\Delta E_{00}))\right)\right|\right] \qquad (15.1) \\
&\leq \frac{\mathcal{K}}{n_j}. \qquad (15.2)
\end{aligned}$$

Inequality (15.1) comes from the fact that T_j and $(\mathbf{x},\mathbf{x}',\Delta E_{00})$ are drawn i.i.d. from the distribution $P(C_j)$ and thus we do not change the expected value by replacing one example with another, (15.2) is obtained by applying triangle inequality followed by the property of uniform stability (Lemma E).

Lemma G *For any matrix $\mathbf{M_j}$ learned by our algorithm using n_j training examples, and any loss function l satisfying the (σ,m)-admissibility, we have*

$$\left|R_{T_j} - R_{T_j^k}\right| \leq \frac{2\mathcal{K} + (\Delta_{\max}\sigma + m)}{n_j}.$$

Proof.

$$\begin{aligned}
\left|R_{T_j} - R_{T_j^i}\right| &= \left|\varepsilon_{P(C_j)}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j}) - \left(\varepsilon_{P(C_j)}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i})\right)\right| \\
&= \left|\varepsilon_{P(C_j)}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j}) - \varepsilon_{P(C_j)}(\mathbf{M_j^i}) + \hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j^i}) + \hat{\varepsilon}_{T_j}(\mathbf{M_j^i})\right| \\
&\leq \left|\varepsilon_{P(C_j)}(\mathbf{M_j}) - \varepsilon_{P(C_j)}(\mathbf{M_j^i})\right| + \left|\hat{\varepsilon}_{T_j}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right| + \left|\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j^i})\right| \qquad (16.1) \\
&\leq \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})}\left[\left|l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00})) - l(\mathbf{M_j^i},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right|\right] + \\
&\quad \left|\hat{\varepsilon}_{T_j}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right| + \left|\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j^i})\right| \qquad (16.2) \\
&\leq \frac{\mathcal{K}}{n_j} + \left|\hat{\varepsilon}_{T_j}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right| + \left|\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j^i})\right| \qquad (16.3) \\
&\leq \frac{\mathcal{K}}{n_j} + \frac{1}{n_j}\sum_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\in T_j} \left|l(\mathbf{M_j^i},(\mathbf{x},\mathbf{x}',\Delta E_{00})) - l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right| + \\
&\quad \left|\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j^i})\right| \\
&\leq \frac{\mathcal{K}}{n_j} + \frac{\mathcal{K}}{n_j} + \left|\hat{\varepsilon}_{T_j^i}(\mathbf{M_j^i}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j^i})\right| \qquad (16.4) \\
&= \frac{2\mathcal{K}}{n_j} + \frac{1}{n_j}\left|l(\mathbf{M_j^i},(\mathbf{t_i},\mathbf{t'_i},\Delta E_{00})) - l(\mathbf{M_j^i},(\mathbf{x_i},\mathbf{x'_i},\Delta E_{00}))\right| \qquad (16.5) \\
&\leq \frac{2\mathcal{K}}{n_j} + \frac{1}{n_j}\left(\sigma\left|\Delta E_{00}\left(\mathbf{t_i},\mathbf{t'_i}\right) - \Delta E_{00}\left(\mathbf{x_i},\mathbf{x'_i}\right)\right| + m\right) \qquad (16.6) \\
&\leq \frac{2\mathcal{K} + (\Delta_{\max}\sigma + m)}{n_j}. \qquad (16.7)
\end{aligned}$$

Inequalities (16.1) and (16.2) are due to the triangle inequality. (16.3) and (16.4) come from the uniform stability (Lemma E). (16.5) comes from the fact that T_j and T_j^i only differ by their i^{th} example. (16.6) comes from the (σ, m)-admissibility of our loss (Lemma C). Noting that $|\Delta E_{00}\left(\mathbf{t_i}, \mathbf{t'_i}\right) - \Delta E_{00}\left(\mathbf{x_i}, \mathbf{x'_i}\right)| \leq \Delta_{\max}$ gives inequality (16.7).

Lemma 1 (Generalization bound) *With probability* $1 - \delta$*, for any matrix* $\mathbf{M_j}$ *related to a region* C_j*,* $0 \leq j \leq K$*, learned with Algorithm 1, we have:*

$$\varepsilon_{P(C_j)}(\mathbf{M_j}) \leq \hat{\varepsilon}_{T_j}(\mathbf{M_j}) + \frac{2D_j^4}{\lambda_j n_j} + \left(\frac{4D_j^4}{\lambda_j} + \Delta_{\max}\left(\frac{2D_j^2}{\sqrt{\lambda_j}} + 2\Delta_{\max}\right)\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n_j}}.$$

Proof. Using the McDiarmid inequality (Theorem A) and Lemma G we can write:

$$\begin{aligned}\Pr\left[\left|R_{T_j} - \mathbb{E}_{T_j}\left[R_{T_j}\right]\right| \geq \epsilon\right] &\leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{j=1}^{n}\left(\frac{2\mathcal{K}+(5\sigma+m)}{n_j}\right)^2}\right) \\ &\leq 2\exp\left(-\frac{2\epsilon^2}{\frac{1}{n_j}\left(2\mathcal{K}+(5\sigma+m)\right)^2}\right).\end{aligned}$$

Then, by setting:

$$\delta = 2\exp\left(-\frac{2\epsilon^2}{\frac{1}{n_j}\left(2\mathcal{K}+(5\sigma+m)\right)^2}\right)$$

we obtain:

$$\epsilon = \left(2\mathcal{K} + (\Delta_{\max}\sigma + m)\right)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}$$

and thus:

$$\Pr\left[\left|R_{T_j} - \mathbb{E}_{T_j}\left[R_{T_j}\right]\right| < \epsilon\right] > 1 - \delta.$$

Then, with probability $1 - \delta$:

$$\begin{aligned} & R_{T_j} < \mathbb{E}_{T_j}\left[R_{T_j}\right] + \epsilon \\ \Leftrightarrow \quad & \varepsilon_{P(C_j)}(\mathbf{M_j}) - \hat{\varepsilon}_{T_j}(\mathbf{M_j}) < \mathbb{E}_{T_j}\left[R_{T_j}\right] + \epsilon \\ \Leftrightarrow \quad & \varepsilon_{P(C_j)}(\mathbf{M_j}) < \hat{\varepsilon}_{T_j}(\mathbf{M_j}) + \frac{\mathcal{K}}{n_j} + \left(2\mathcal{K} + (\Delta_{\max}\sigma + m)\right)\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n_j}}. \end{aligned}$$

The last equation is obtained by using Lemma F and replacing $\mathcal{K}$, σ and m by their respective values gives the lemma.

We showed that our approach is locally consistent. In the next section, we show that our algorithm globally converges in $O(1/\sqrt{n})$.

2.2 Generalization bound for Algorithm 1

We consider the partition $C_0, C_1, \ldots, C_K$ over pairs of examples considered by Algorithm 1. We first recall the concentration inequality that will help us to derive the bound.

Proposition 1 ([3]). *Let $(n_0, n_1, \ldots, n_K)$ an IID multinomial random variable with parameters $n = \sum_{j=0}^{K} n_j$ and $(P(C_0), P(C_1), \ldots, P(C_K))$. By the Breteganolle-Huber-Carol inequality we have: $Pr\left\{\sum_{j=0}^{K} \left|\frac{n_j}{n} - P(C_j)\right| \geq \eta\right\} \leq 2^K \exp\left(\frac{-n\eta^2}{2}\right)$, hence with probability at least $1-\delta$,*

$$\sum_{j=0}^{K} \left|\frac{n_j}{n} - P(C_j)\right| \leq \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}. \tag{17}$$

We recall the true and empirical risks. Let $\mathbf{M} = \{\mathbf{M_0}, \mathbf{M_1}, \ldots, \mathbf{M_K}\}$ be the $K+1$ matrices learned by our algorithm. The true error associated to $\mathbf{M}$ is defined as $\varepsilon(\mathbf{M}) = \sum_{j=0}^{K} \varepsilon_{P(C_j)}(\mathbf{M_j}) P(C_j)$ where $\varepsilon_{P(C_j)}(\mathbf{M_j})$ is the local true risk for C_j. The empirical error over T of size n is defined as $\hat{\varepsilon}_T(\mathbf{M}) = \frac{1}{n}\sum_{j=0}^{K} n_j \hat{\varepsilon}_{T_j}(\mathbf{M_j})$ where $\hat{\varepsilon}_{T_j}(\mathbf{M_j})$ is the empirical risk of T_j.

Before proving the main theorem of the paper we introduce an additional lemma showing a bound on the loss function.

Lemma H *Let $\mathbf{M} = \{\mathbf{M_0}, \mathbf{M_1}, \ldots, \mathbf{M_K}\}$ be any set of metrics learned by Algorithm 1 from a data sample T of n pairs, for any $0 \leq j \leq K$, we have that for any example $(\mathbf{x}, \mathbf{x}', \Delta E_{00}) \sim P(C_j)$:*

$$l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) \leq L_B,$$

with $L_B = \max\{\frac{\Delta_{max}}{\sqrt{\lambda}}, \Delta^2_{\max}\}$.

Proof.

$$\begin{aligned}
l(\mathbf{M_j}, (\mathbf{x}, \mathbf{x}', \Delta E_{00})) &= \left|(\mathbf{x}-\mathbf{x}')^T \mathbf{M_j} (\mathbf{x}-\mathbf{x}') - \Delta E_{00}\left(\mathbf{x}, \mathbf{x}'\right)^2\right| \\
&\leq \max\left\{(\mathbf{x}-\mathbf{x}')^T \mathbf{M_j} (\mathbf{x}-\mathbf{x}'), \Delta E_{00}\left(\mathbf{x}, \mathbf{x}'\right)^2\right\} && (18.1)\\
&\leq \max\left\{\frac{\Delta_{\max}}{\sqrt{\lambda}}, \Delta E_{00}\left(\mathbf{x}, \mathbf{x}'\right)^2\right\} && (18.2)\\
&\leq \max\left\{\frac{\Delta_{\max}}{\sqrt{\lambda}}, \Delta^2_{\max}\right\}. && (18.3)
\end{aligned}$$

Inequality (18.1) comes from the fact that any matrix $\mathbf{M_j}$ is positive semi definite and thus we are taking the absolute difference of two positive values. Inequality (18.2) is obtained by using the Cauchy-Schwarz inequality, the Lemma B with $\lambda = \min_{0\leq j\leq K} \lambda_j$ and the inequality $\|\mathbf{x}-\mathbf{x}'\| \leq 1$. Inequality (18.3) is due to the definition of $\Delta_{\max}$.

We can now prove the main theorem of the paper.

Theorem 1 *Let $C_0, C_1, \ldots, C_K$ be the regions considered and $\mathbf{M} = \{\mathbf{M_0}, \mathbf{M_1}, \ldots, \mathbf{M_K}\}$ any set of metrics learned by Algorithm 1 from a data sample T of n pairs, we have with probability at least $1-\delta$ that*

$$\begin{aligned}
\varepsilon(\mathbf{M}) \leq & \hat{\varepsilon}_T(\mathbf{M}) + L_B\sqrt{\frac{2(K+1)\ln 2 + 2\ln(2/\delta)}{n}} + \frac{2(KD^4+1)}{\lambda n} \\
& + \left(\frac{4(KD^4+1)}{\lambda} + \Delta_{\max}\left(\frac{2(KD^2+1)}{\sqrt{\lambda}} + 2(K+1)\Delta_{\max}\right)\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n}}
\end{aligned}$$

where $D = \max_{1\leq j\leq K} D_j$, L_B is the bound on the loss function and $\lambda = \min_{0\leq j\leq K}\lambda_j$ is the minimum regularization parameter among the $K+1$ learning problems used in Algorithm 1.

Proof. Let n_j be the number points of T that fall into the partition C_j. $(n_0, n_1, \ldots, n_K)$ is a IID multinomial random variable with parameters n and $(P(C_0), P(C_1), \ldots, P(C_K))$.

$$
\begin{aligned}
|\varepsilon(\mathbf{M}) - \hat{\varepsilon}_T(\mathbf{M})| &= \left|\mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P}\left[l(\mathbf{M},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] - \hat{\varepsilon}_T(\mathbf{M})\right| \\
&= \left|\sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] P(C_j) - \hat{\varepsilon}_T(\mathbf{M})\right| \\
&= \Bigg|\sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] P(C_j) \\
&\quad - \sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] \frac{n_j}{n} \\
&\quad + \sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] \frac{n_j}{n} - \hat{\varepsilon}_T(\mathbf{M})\Bigg| \\
&\le \Bigg|\sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] P(C_j) \\
&\quad - \sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] \frac{n_j}{n}\Bigg| \\
&\quad + \left|\sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] \frac{n_j}{n} - \hat{\varepsilon}_T(\mathbf{M})\right| \qquad (19.1) \\
&\le \sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left|\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right]\right| \left|P(C_j) - \frac{n_j}{n}\right| \\
&\quad + \left|\sum_{j=0}^{K} \mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] \frac{n_j}{n} - \sum_{j=0}^{K} \frac{n_j}{n}\hat{\varepsilon}_{T_j}(\mathbf{M_j})\right| \qquad (19.2) \\
&\le \sum_{j=0}^{K} L_B \left|P(C_j) - \frac{n_j}{n}\right| \\
&\quad + \left|\sum_{j=0}^{K} \frac{n_j}{n}\left(\mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right)\right| \qquad (19.3) \\
&\le L_B\sqrt{\frac{2(K+1)\ln 2 + 2\ln(2/\delta)}{n}} \\
&\quad + \sum_{j=0}^{K} \frac{n_j}{n}\left|\mathbb{E}_{(\mathbf{x},\mathbf{x}',\Delta E_{00})\sim P|(\mathbf{x},\mathbf{x}',\Delta E_{00})\in C_j}\left[l(\mathbf{M_j},(\mathbf{x},\mathbf{x}',\Delta E_{00}))\right] - \hat{\varepsilon}_{T_j}(\mathbf{M_j})\right| \qquad (19.4) \\
&\le L_B\sqrt{\frac{2(K+1)\ln 2 + 2\ln(2/\delta)}{n}} \\
&\quad + \sum_{j=0}^{K} \frac{n_j}{n}\left(\frac{2D_j^4}{\lambda_j n_j} + \left(\frac{2D_j^4}{\lambda_j} + \Delta_{\max}\left(\frac{2D_j^2}{\sqrt{\lambda_j}} + 2\Delta_{\max}\right)\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n_j}}\right) \qquad (19.5) \\
&\le L_B\sqrt{\frac{2(K+1)\ln 2 + 2\ln(2/\delta)}{n}} + \frac{2(KD^4+1)}{\lambda n} \\
&\quad + \left(\frac{2(KD^4+1)}{\lambda} + \Delta_{\max}\left(\frac{2(KD^2+1)}{\sqrt{\lambda}} + 2\Delta_{\max}\right)\right)\sqrt{\frac{\ln(\frac{4(K+1)}{\delta})}{2n}} \qquad (19.6)
\end{aligned}
$$

Inequalities (19.1) and (19.2) are due to the triangle inequality. (19.3) comes from the application of Lemma H. Inequality (19.4) is obtained by applying Proposition 1 with probability $1 - \delta/2$. (19.5) is due to the application of Lemma 1 with probability $1 - \delta/(2(K+1))$ for each of the $(K+1)$ learning problems. Inequality (19.6) is obtained by cancelling out the n_j, noting that $\sqrt{n_j} \leq \sqrt{n}$ and taking $D = max_{1 \leq i \leq n} D_j$. Note that $D_0 = 1$ corresponds to the partition used by the global metric.

Eventually by the union bound we obtained the final result with probability $1 - \delta$.

3 Image Segmentation

In this section, we illustrate the application of the color mean-shift algorithm presented in our paper. We apply color mean-shift on RGB components, on $L^*u^*v^*$ components and by using our learned distance directly in the RGB components. The overall quantitative results for the Berkeley dataset are provided in the paper and we propose to show some qualitative results on this dataset in Figure 1. As explained in the paper, the number of segments in the resulting images is not a parameter of the algorithm, as a consequence it is not easy to obtain images with the same number of segments for the three algorithms (RGB, $L^*u^*v^*$ and Metric learning). Thus, given an image, by playing with the color distance threshold, we have tried to obtain the same segment numbers as the corresponding ground truth for the three algorithms. However, the color mean-shift algorithm provides some very small segments, specially for the RGB and $L^*u^*v^*$ color spaces. Consequently, for each test, in Figure 1, we have mentioned between brackets, first, the number of segments, and second, the number of segments whose size is more than 150 pixels. For a fair comparison, we use this last number as reference for each image, i.e. this number is almost constant and close to the ground truth for each row.

It is worth mentioning that the ground truth segmentation has always very few segments. Thus, starting from a large number of small segments, the used algorithm is grouping them by considering their color differences. Consequently, the used color distance is crucial when we want to obtain small number of segments as provided by the ground truth. We can see in Figure 1, that when working in the RGB or $L^*u^*v^*$ color spaces, some segments that are perceptually different are merged while some other similar segments are not. Most of the time, the color mean-shift is working well when using our distance. This point was already checked quantitatively on the whole Berkeley dataset in the paper.

References

1. Olivier Bousquet and André Elisseeff. Stability and generalization. *JMLR*, 2:499–526, 2002.
2. Colin McDiarmid. *Surveys in Combinatorics*, chapter On the method of bounded differences, pages 148–188. Cambridge University Press, 1989.
3. Aad W. van der Vaart and Jon A. Wellner. *Weak convergence and empirical processes*. Springer, 2000.

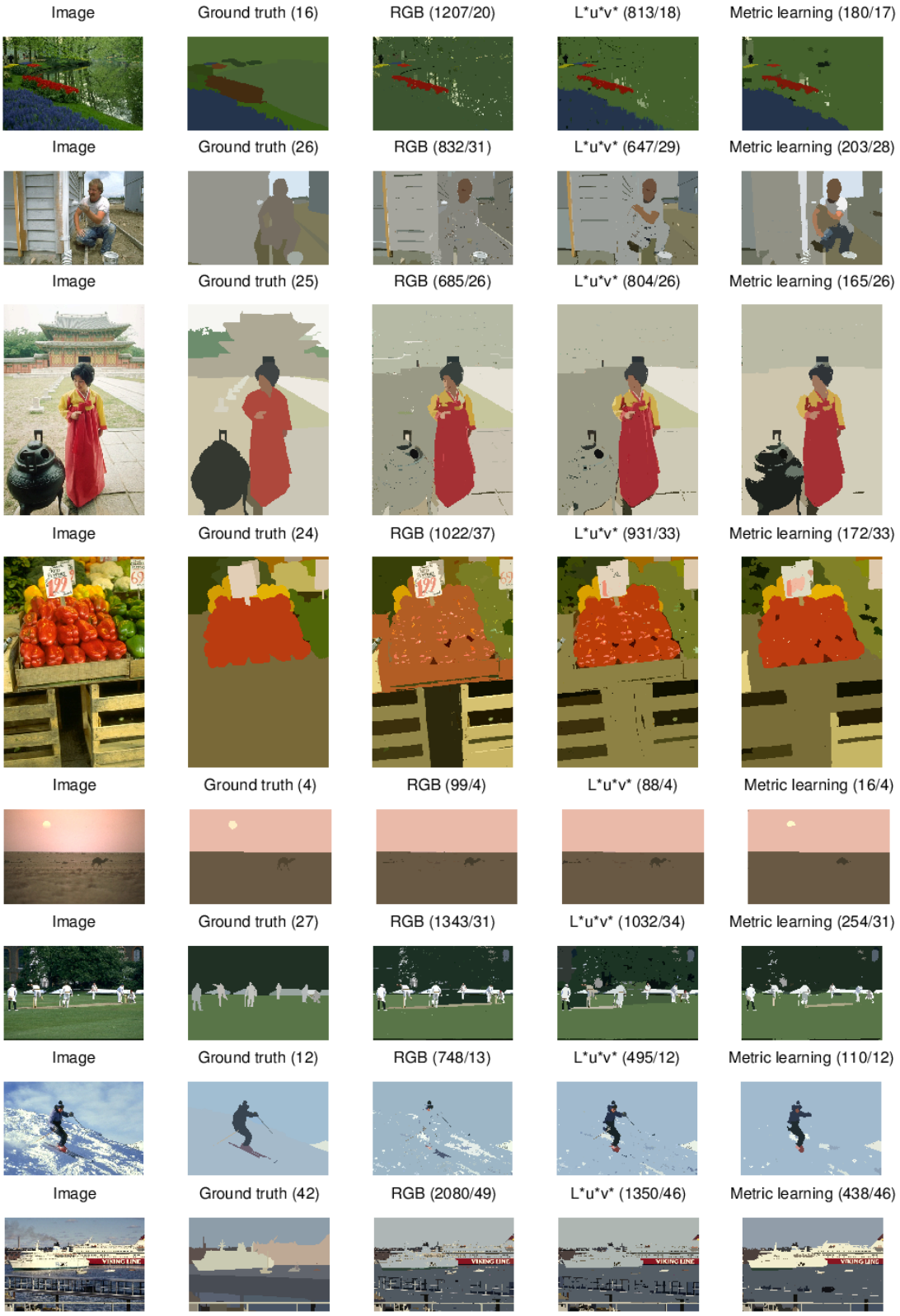


Fig. 1. Illustration of segmentation provided by the color mean-shift algorithm applied in the RGB components (third column), on $L^*u^*v^*$ components (fourth column) and by using our learned distance directly in the RGB components (fifth column). First column represents the original image and the second one the ground truth.